

Software–Projekt 2 2013/2014

VAK 03-BA-901.02

Projektplan

Sebastian Bredehöft	sbrede@tzi.de	2751589
Patrick Damrow	damsen@tzi.de	2056170
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Mohamadreza (Amir) Khostevan	amirkh@tzi.de	1234567

Inhaltsverzeichnis

1	Einleitung	4
1.1	Projektübersicht	4
1.1.1	Ziele	4
1.1.2	Hauptarbeitsaktivitäten und -produkte	5
1.1.3	Haupt-Meilensteine und grober Zeitplan	5
1.1.4	Benötigte Ressourcen	6
1.1.5	Budget	7
1.1.6	Kontaktdaten des Kunden	7
1.1.7	Mitarbeiter	7
1.2	Auszuliefernde Produkte	8
1.3	Evolution des Plans	8
1.4	Referenzen	8
1.5	Definitionen und Akronyme	8
2	Projektorganisation	8
2.1	Prozessmodell	8
2.2	Organisationsstruktur	8
2.3	Organisationsgrenzen und -schnittstellen	8
2.4	Verantwortlichkeiten	8
3	Managementprozess	9
3.1	Managementprozess und -prioritäten	9
3.2	Annahmen, Abhängigkeiten und Einschränkungen	10
3.2.1	Annahmen	10
3.2.2	Abhängigkeiten	11
3.2.3	Einschränkungen	11
3.3	Risikomanagement	11
3.4	Projektüberwachung	13
3.5	Mitarbeiter	13
4	Technische Prozesse	13
4.1	Methoden, Werkzeuge und Techniken	13
4.1.1	Entwicklungsplattform	13
4.1.2	Entwicklungsmethode	14
4.1.3	Programmiersprache und Bibliotheken	14
4.2	Dokumentationsplan	14
4.2.1	Codingstyle	14
4.2.2	Kommentarsprache	15
4.2.3	JavaDoc	15
4.2.4	Begleitende Dokumentation	15
4.3	Unterstützende Projektfunktionen	15

5	Arbeitspakete, Zeitplan und Budget	15
5.1	Arbeitspakete	16
5.2	Zeitplan und Abhängigkeiten	17
5.3	Ressourcenanforderung	17
6	Sonstige Elemente	17
6.1	Pläne für die Konvertierung von Daten	17
6.2	Managementpläne für Unterauftragsnehmer	17
6.3	Ausbildungspläne	17
6.4	Raumpläne	17
6.5	Installationspläne	17
6.6	Pläne für die Übergabe des Systems	18

Version und Änderungsgeschichte

Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.

Version	Datum	Änderungen
0.1	14.10.2013	Ziele hinzugefügt.
0.1.1	14.10.2013	Ziele vervollständigt
0.2	15.10.2013	Hauptarbeitsaktivitäten und -produkte hinzugefügt.
0.3	15.10.2013	Meilensteine eingefügt.
0.4	15.10.2013	Benötigte Ressourcen -Menschen hinzugefügt.
0.5	16.10.2013	Ressourcen ergänzt -Hardware und -Räume.
1.0	20.10.2013	Erste veröffentlichte Version.

1 Einleitung

Dieses Dokument ist der Projektplan der Gruppe IT_R3V0LUT10N im des Softwareprojekts im Wintersemester 2013/2014 an der Universität Bremen. Der Projektplan entspricht der Struktur ANSI/IEEE Std. 1058.1-1987¹.

1.1 Projektübersicht

1.1.1 Ziele

Das Ziel unserer Gruppe IT_R3V0LUT10N ist es, das Softwareprojekt 2 der Universität Bremen zu bestehen. Dies setzt die Einhaltung der Fristen und Termine, eine ausreichende Fertigstellung des Projekts und die Abgabe aller in SWP2 geforderten Dokumente wie Projektplan, Anforderungsspezifikation und Angebot, Architekturbeschreibung, Schnittstellenbeschreibung, Testplan inklusive Blackbox-Tests und ein elektronisch geführtes Berichtsheft voraus. Darüber hinaus wollen wir einen GUI-Prototypen erstellen und den Akzeptanztest bestehen. Ein Bibliothekssystem zu erstellen steht aber im Vordergrund.

Das Bibliothekssystem beinhaltet sowohl eine Website, als auch einen Zugang für mobile Geräte mit kleinem Display. Ziel ist es, die Mindestanforderungen² und eventuell weitergehende Funktionen zu implementieren.

Zu den Mindestanforderungen gehören die Erstellung und Abgabe einer Bibliothekssoftware, eines Serverprogramms mit Datenbankanbindung, einen Administrationszugang und einen Zugang für mobile Geräte mit kleinem Display. Wir haben uns entschieden den Zugang für die mobilen Geräte in Form einer Android-App zu realisieren, weil wir

¹<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=25325&userType=inst>

²http://www.informatik.uni-bremen.de/st/Lehre/swpII_1314/mindestanforderungen.html

dies für Zeitgemäß und einfacher für den Leser, an den diese Form des Zugangs gerichtet ist, halten. Die zu erstellende Bibliothekssoftware dient in erster Linie zur Verwaltung des Medienbestandes der Bibliothek und dem Verleihen dieser Medien an der Oberschule Rockwinkel. Der Administrationszugang wird benötigt um Bibliothekare anzulegen, zu löschen, deren Stammdaten zu ändern, oder deren Rechte innerhalb der Software zu verändern.

(TODO: Weitere Features sind den oben genannten Mindestanforderungen zu entnehmen, oder werden im weiteren Verlauf des Projektplans erläutert.)

1.1.2 Hauptarbeitsaktivitäten und -produkte

(TODO: Beschreibung)

Tabelle 1: Hauptaktivitäten und -produkte

Aktivität	Arbeitsprodukt
Projektplanung	Projektplan
Anforderungsanalyse, Angebotserstellung	Anforderungsspezifikation, Angebot
Entwurf (Globale Analyse, Konzeptionelles Modell, Modulblickwinkel, Ausführungsblickwinkel, Codeblickwinkel)	Architekturbeschreibung
erstellen des Testplans, Tests	Testplan, Schnittstellentests
Implementierung	lauffähiges Programm
Dokumentation	Installationsanweisung/-Skript
Auslieferung	Kunde erhält Produkt

1.1.3 Haupt-Meilensteine und grober Zeitplan

M0 - 14.10.2013 Beginn des Projektes

M1 - 20.10.2013 Abgabe initialer Projektplan

Jedes Mitglied muss seinen Teil fertig gestellt haben. Anschließend werden alle Einzelteile zusammengeführt und von allen auf Korrektheit geprüft.

M2 - 13.11.2013 Anforderungsspezifikation (Intern)

Jedes Mitglied hat seinen Teil der Anforderungsspezifikation fertiggestellt. Anschließend werden die Teile zusammengeführt und von allen auf Korrektheit geprüft.

M3 - 17.11.2013 Abgabe der Anforderungsspezifikation

Meilenstein 2 muss bereits fertig sein. Abgabe der Anforderungsspezifikation via MEMS.

M4 - 18.12.2013 Architektur- und Schnittstellenbeschreibung, Testplan, Tests (Intern)
Jedes Mitglied muss seine Aufgaben erfüllt haben. Teile werden zusammengeführt und kontrolliert. Tests müssen implementiert sein.

M5 - 22.12.2013 Architekturbeschreibung, Testplan und Schnittstellentests fertig
Meilenstein 4 muss bereits erreicht worden sein. Tests wurden lauffähig implementiert. Abgabe via MEMS.

M6 - 26.01.2014 Erste lauffähige Basisversion
Jedes Mitglied muss seine Arbeitspakete fertig gestellt haben. Das Team muss die oben genannten Mindestanforderungen (**TODO: referenz auf fußnote**), die mit einem * markiert sind, implementiert haben

M7 - 23.02.2014 Vollständige Abgabe der Dokumente und der Software
Die Software muss lauffähig und vollständig implementiert sein,
Abgabe des Build-/Installationsskriptes

1.1.4 Benötigte Ressourcen

- **Menschliche Ressourcen**

An Menschlichen Ressourcen stehen sechs Informatik Studenten der Universität Bremen zur Verfügung. Wir haben als durchschnittliche Arbeitszeit pro Woche und Person einen Aufwand von ca. 14,5 Stunden für das Projekt errechnet. Dieser Wert ergibt sich folgendermaßen:

Für das Modul Software Projekt 2 gibt es 9CP. 1CP entspricht 30 Semesterstunden. $9 \times 30 = 270$ Stunden. Da wir 19 Wochen lang an dem Projekt arbeiten werden, ergibt sich ein aufgerundeter Wert von 14,5 Stunden pro Woche ($270 / 19 = 14,21$). Unsere Kontaktdaten sind dem Punkt Mitarbeiter zu entnehmen. (**TODO: Referenz**)

- **Hard-/ und Software**

Jedes unserer Mitglieder ist im Besitz, oder hat Zugriff, auf Computer, die folgenden Anforderungen und Verfügbarkeiten gerecht werden müssen:

- zum Anfertigen der Dokumente wird ein Textsatzprogramm benötigt (L^AT_EX wird bevorzugt).
- für die Entwicklung der Software müssen Java-Runtime, ein Texteditor und eine Entwicklungsumgebung mit Android-SDK installiert sein.
- Git wird zum gleichzeitigen Bearbeiten der Dokumente und zum Datenaustausch der Entwickler benötigt.

- **Räume**

Das Team wird sich während der gesamten Projektlaufzeit Montags, soweit verfügbar, in einer der Lerninseln im GW2 A2370 oder A3440 der Universität Bremen von 10 Uhr bis 14 Uhr treffen. Weitere spezielle Räumlichkeiten werden

nicht benötigt, da wir den Kontakt regelmäßig via Skype oder Email gewährleisten.

1.1.5 Budget

Ein Budget für dieses Projekt in Form von Geld entfällt, da die Software im Rahmen des Moduls Software Projekt 2 entwickelt wird. Wenn wir über 19 Wochen (vom 14.10.2013 bis zum 23.02.2014) an dem Projekt mit 6 Studenten 14,5 Stunden pro Woche arbeiten, ergibt sich eine Gesamtsumme von 1653 Entwicklerstunden ($19 \times 6 \times 14,5 = 1653$).

Wir entnehmen einer Studie von Gulp ³ das zwei Drittel der Software-Entwickler zwischen 60 und 80 Euro fordern. Da wir alle Studenten sind und somit noch in der Ausbildung, setzen wir den Studienlohn für jeden Entwickler bei 40 Euro an. Somit würden sich für den Arbeitsaufwand der Entwicklerstunden Kosten von insgesamt 66.120 Euro ergeben.

1.1.6 Kontaktdaten des Kunden

*Oberschule Rockwinkel
Uppe Angst 31
28355 Bremen
Telefon: 0421 - 361 16 627
Fax: 0421 - 361 16 637
E-Mail: 416@bildung.bremen.de*

1.1.7 Mitarbeiter

Name	Email	Foto
Bredehöft, Sebastian	sbrede@tzi.de	bild einfügen
Damrow, Patrick	damsen@tzi.de	bild einfügen
Dellert, Tobias	tode@tzi.de	bild einfügen
Pupat, Daniel	dpupat@informatik.uni-bremen.de	bild einfügen
Ellhoff, Tim	tellhoff@tzi.de	bild einfügen
Khostevan, Mohamadreza (Amir)	amirkh@tzi.de	bild einfügen

³<http://www.gulp.de/presse/pressemitteilungen/marktstudie-freiberufliche-software-entwickler-sind.html>

1.2 Auszuliefernde Produkte

1.3 Evolution des Plans

ENTFÄLLT *Wird der Plan verändert? Wann? Wie oft? Von wem? Wenn bereits Aktualisierungen vorgesehen sind, welche sind das? Möglicherweise betrifft das die Zeitplanung, die Risikobewertung, oder andere Teile des Plans. Gibt es möglicherweise auch unvorhergesehene Aktualisierungen?*

1.4 Referenzen

1.5 Definitionen und Akronyme

ENTFÄLLT *Hier sollen Begriffe definiert werden, die nötig sind, um den Projektplan zu verstehen. Diese kommen insbesondere aus der Welt des Kunden (Projektdomäne) und der Welt des Softwareproduzenten.*

2 Projektorganisation

ENTFÄLLT

2.1 Prozessmodell

ENTFÄLLT

2.2 Organisationsstruktur

ENTFÄLLT *Genaue Beschreibung der Rollen, Rechte und Pflichten!*

z.B. auch regelmäßiges Treffen im Chat, Einrichtung einer Groupware oder eines Forums, o.ä. ...

2.3 Organisationsgrenzen und –schnittstellen

ENTFÄLLT *Hierher gehören auch evtl. Kontaktpersonen für Fremdbibliotheken u.ä.*

2.4 Verantwortlichkeiten

ENTFÄLLT

3 Managementprozess

3.1 Managementprozess und –prioritäten

Folgende Managementprozesse haben bei uns die höchsten Prioritäten:

Fertigstellung des Produktes:

Ein Ziel von uns ist die Fertigstellung des Produktes, welches vom Kunden gefordert ist. Dabei ist wichtig alle Mindestanforderungen, die der Kunde gefordert hat, erfolgreich umgesetzt wurden.

Dieses Ziel hat die höchste Priorität, da dies notwendig ist, um die Veranstaltung SWP 2 zu bestehen und eine Vorraussetzung aller anderen aufgeführten Ziele ist.

Qualität des Produktes:

Ein weiteres Ziel ist es, dem Produkt eine hohe Qualität zu geben. Dies ist notwendig, damit der Kunde zufrieden ist und das Produkt später evtl. verwendet wird. Dabei ist wichtig, das neben den Mindestanforderungen, weitere Funktionen vorhanden sind und die Benutzung einfach und benutzerfreundlich ist.

Dieses Ziel hat eine hohe Priorität, da dies notwendig ist um den Kunden zufrieden zu stellen und eine gute Note zu erreichen.

Weiterentwicklung des Produktes:

Es ist auch wichtig das Produkt so zu entwickeln, damit dieses später bei Bedarf von anderen weiterentwickelt werden kann. Dies erfordert eine Strukturierte Implementierung. Dieses Ziel hat niedrige Priorität, da wir in erster Linie das Modul bestehen wollen

Kundenzufriedenheit:

Es ist sehr wichtig das der Kunde später zufrieden ist, was bedeutet, dass man die Mindestanforderungen erfüllt und darüber hinaus noch weitere Features einbindet, da nur so der Kunde wirklich zufrieden ist.

Dieses Ziel hat für uns mittlere Priorität, da wir in erster Linie die Mindestanforderungen schaffen wollen und nur wenn noch Zeit ist weitere Features einbinden. Dieses könnte aber noch notwendig sein um eine gute Note zu erreichen.

Kommunikation innerhalb der Gruppe:

Ein wichtiger Faktor ist noch die Kommunikation innerhalb der Gruppe. Wenn man sich nicht abspricht, kann es zu Schwierigkeiten kommen, wenn z.B. ein Gruppenmitglied seinen Teil nicht rechtzeitig schafft und die anderen aber davon ausgehen.

Dieses Ziel hat bei uns eine hohe Priorität, da ohne Kommunikation das Projekt mit hoher Wahrscheinlichkeit scheitert.

Klima innerhalb der Gruppe:

Ein gutes Gruppenklima heißt, dass innerhalb der Gruppe alle gut miteinander auskommen und es keinen Streit gibt. Außerdem muss man den anderen Vertrauen können, dass sie immer rechtzeitig fertig werden und bei Problemen Bescheid geben.

Dies hat ebenfalls eine hohe Priorität, da gerade das Vertrauen und die Zuverlässigkeit sehr wichtig sind, damit alles rechtzeitig fertig wird.

Gute Note:

Ziel dieser Veranstaltung ist für uns das Projekt so gut wie möglich zu bestehen. Dabei sollte jeder sein bestes geben, damit am Ende das Maximum an Punkten für die Gruppe erreicht wird.

Dies hat bei uns eine hohe Priorität, da wir später einen möglichst guten Abschluss haben wollen.

Kunde entscheidet sich für unser Produkt:

Da der Kunde am Ende der Veranstaltung ein Produkt aussuchen wird, welches dann in der Bibliothek verwendet wird, wäre es möglich das er unser Produkt wählt.

Dieses Ziel hat bei uns eine niedrige Priorität, da wir in erster Linie gut abschneiden wollen, aber nicht darauf hinarbeiten, unbedingt das beste Produkt der Veranstaltung zu entwickeln, da dies zu zeitaufwendig wäre.

3.2 Annahmen, Abhängigkeiten und Einschränkungen

3.2.1 Annahmen

Mindestanforderungen werden nicht verändert:

Die erste Annahme ist, dass der Kunde die Mindestanforderungen nicht verändert. Dies bedeutet, dass es keine Möglichkeit gibt andere Mindestanforderungen auszuhandeln und der Kunde auch keine neuen stellt.

Deadline wird nicht verschoben:

Noch eine Annahme ist, dass sich die Deadlines der verschiedenen Abgaben sich unter normalen Umständen nicht verändern. Dies bedeutet, dass der Kunde diese nicht vorverlegt und wir diese nicht nach hinten verlegen können.

Erfolgreiche Teilnahme:

Eine weitere Annahme ist, dass alle Gruppenmitglieder die Veranstaltung erfolgreich bestehen wollen. Da sich alle für dieses Modul eingetragen haben, kann man davon ausgehen, dass alle ihr bestes geben um diese Veranstaltung zu bestehen.

Grundkenntnisse in Java:

Man kann auch annehmen, dass alle Mitglieder Grundkenntnisse in Java haben, da alle Gruppenmitglieder bereits Praktische Informatik 1 und 2 besucht haben.

3.2.2 Abhängigkeiten

Laptop:

Da jeder von uns ein Laptop besitzt, werden wir diesen hauptsächlich verwenden, da so jeder mobil ist und überall weiterarbeiten kann.

GitHub:

Zum Teilen der Dokumente verwenden wir GitHub. So kann jede Person einen Teil bearbeiten und die Dokumente können dann zusammengeführt werden.

Glassfish:

Als Server verwenden wir den Glassfish Server, auf dem unser Programm später arbeitet.

Mitglieder:

Da dies eine Gruppenarbeit ist, muss jedes Gruppenmitglied ihren Teil leisten, da die Arbeit auf 6 Leute ausgelegt ist.

Von den eben genannten Punkten ist das Projekt abhängig, da bei einem Ausfall der Punkte Schwierigkeiten auftreten können.

3.2.3 Einschränkungen

Weitere belegte Module:

Jeder von uns belegt noch weitere Module und hat deswegen nur eine gewisse Zeit für SWP 2. Hinzu kommt noch, dass wir Mitglieder haben, die in unterschiedlichen Semestern sind, wodurch es schwierig ist einen gemeinsamen Termin zu finden.

3.3 Risikomanagement

Wenn Ihr Euch entschieden habt, bestimmte vorbeugende Maßnahmen durchzuführen, solltet Ihr dies deutlich kennzeichnen. Hoffentlich haben diese Maßnahmen dann einen Einfluss auf Eintrittswahrscheinlichkeit oder Schadenshöhe (zum Beispiel ist die Eintrittswahrscheinlichkeit von komplettem Datenverlust durch regelmäßige Backups deutlich geringer). Daher solltet Ihr für diese Fälle dann die verringerten Werte für Eintrittswahrscheinlichkeit, Schadenshöhe und Risikopotential zusätzlich angeben.

Wie werden neue Risiken erkannt/erfasst? Wer ist für was zuständig? Wie ist der Informationsfluss? ...

Dieser Teil ist ein wichtiger Schwerpunkt des Projektplans und sollte daher ausführlich behandelt werden.

Risiko	EW (1-10)	SH (1-10)	RH
Krankheitsbedingter Ausfall eines Gruppenmitglieds	5	4	20
Krankheitsbedingter Ausfall mehrerer Gruppenmitglieder	2	7	14
Austritt eines Gruppenmitglieds	4	5	20
Austritt mehrerer Gruppenmitglieder	1	8	8
Inkompetenz eines Gruppenmitglieds	2	7	14
Mangelhafte Kommunikation innerhalb der Gruppe	4	6	24
Auflösung/Teilung der Gruppe	2	10	20
Unstimmigkeiten in der Gruppe	2	5	10
Mangelnde Motivation in der Gruppe	6	5	30
Zeitmangel	6	6	36
Probleme mit neuen Technologien	5	3	15
Ausfall von GitHub	1	8	8
Ausfall des Glassfish-servers	1	8	8

EW = Eintrittswahrscheinlichkeit (Skala 1:gering - 10:hoch)

SH = Schadenshöhe (Skala 1:gering - 10:hoch)

RH = Risikohöhe (EW * SH)

Krankheitsbedingter Ausfall eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Krankheiten fallen eine oder mehrere Personen aus und können nicht mehr richtig oder für eine gewisse Zeit überhaupt nicht mehr mitarbeiten. Dadurch kommt auf die restliche Gruppe mehr Arbeit zu.

Maßnahmen:

1. Gruppenmitglied benachrichtigt die anderen Mitglieder so früh wie möglich, damit diese sich darauf einstellen können.
2. Die Gruppe sucht Gespräch mit dem Tutor wenn mehrere Personen ausfallen.

Austritt eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Zeitmangel, Studienabbruch und anderen Gründen kann es jederzeit passieren, dass Gruppenmitglieder aus der Gruppe austreten. Dadurch müssen die anderen Personen dann entsprechend mehr arbeiten, was zu Problemen führen kann.

Maßnahmen:

1. Bei einem Austritt aus der Gruppe gibt das Mitglied den anderen sofort Bescheid, damit diese sich rechtzeitig auf die Mehrarbeit einstellen können.

2. Sollten mehrere Mitglieder austreten, Gespräch mit dem Tutor suchen um gegebenenfalls die Anforderungen zu senken.
3. Im Zeitplan vor den Deadlines immer ein wenig Zeit überlassen, um durch einen plötzlichen Austritt die Abgabe noch rechtzeitig zu schaffen.

Inkompetenz eines Gruppenmitglieds:

Es kann passieren das ein Gruppenmitglied Inkompetent ist und somit nicht in der Lage die ihm zugeteilten Aufgaben zu lösen. Dabei kann es passieren, dass

3.4 Projektüberwachung

Um das Projekt zu überwachen wird mindestens einmal die Woche ein Treffen stattfinden, wo überprüft wird, wer wie weit ist. Außerdem wird es einen permanenten Austausch über Skype geben, wo nachgefragt wird, ob es noch Probleme gibt.

Es wird auch für jede Phase einen Phasenleiter geben, der dafür zuständig ist, den Zeitplan im Auge zu behalten. Diesem Phasenleiter muss dann jedes Gruppenmitglied regelmäßig Bescheid geben, wie weit die Teilaufgabe bereits bearbeitet ist.

Sollte es Probleme bzw. Verzögerungen geben, werden diese im wöchentlichen treffen angesprochen und gemeinsam gelöst, indem z.B die in Verzögerung geratene Aufgabe unter den Mitgliedern aufgeteilt wird.

Bei größeren Probleme kann ein Treffen spontan einberufen werden oder es wird direkt bei Skype angesprochen und da versucht zu lösen. Dabei ist wichtig, dass die Probleme bzw. Verzögerungen immer von der Gruppe erledigt werden, damit es nicht zur Verärgerung kommt, wenn der Phasenleiter dies alleine entscheidet.

3.5 Mitarbeiter

Kompetenzen der und Anforderungen an die Mitarbeiter.

4 Technische Prozesse

4.1 Methoden, Werkzeuge und Techniken

4.1.1 Entwicklungsplattform

Folgende Werkzeuge werden im Entwicklungsprozess von uns benutzt:

- Eclipse⁴ ist unsere Entwicklungsumgebung (beinhaltet AndroidSDK⁵ für die Androidentwicklung

⁴<http://www.eclipse.org/>

⁵<https://developer.android.com/sdk/index.html>

- Maven⁶ ist unser Build-Management Tool
- GlassFish 3.1⁷ ist unser Application-Server
- jUnit⁸ ist unser Framework zum testen
- GitHub⁹ zur Versionsverwaltung
- GanttProject¹⁰ für Gantt-Diagramme
- L^AT_EX¹¹ zur Dokumentenerstellung

(TODO: Sind das alle?)

4.1.2 Entwicklungsmethode

(TODO: Abhängigkeit von Prozessmodell, Mittwoch abstimmen)

4.1.3 Programmiersprache und Bibliotheken

Die Programmiersprache wird Java (mindestens Version 5) sein. Außerdem wird in geringen Umfang HTML und XML benutzt.

Ob und welche Bibliotheken genutzt werden, kann zu diesem Zeitpunkt (*Abgabe: 20.10.2013*) nicht gesagt werden.

(TODO: Android SDK)

Sobald wir Bibliotheken nutzen, wird dieser Punkt aktualisiert.

4.2 Dokumentationsplan

Wir werden als Ergebnis verschiedene Dokumentationen vorweisen können. Diese sind:

- Nutzerhandbuch
- Installationsanleitung
- Dokumentation des Quellcodes

4.2.1 Codingstyle

Unsere Implementierungen werden sich an die *Code Conventions for the Java Programming Language*¹² halten.

⁶<http://maven.apache.org/>

⁷<http://glassfish.java.net/>

⁸<http://junit.org/>

⁹<http://github.com/>

¹⁰<http://www.ganttproject.biz/>

¹¹<http://www.latex-project.org/>

¹²<http://www.oracle.com/technetwork/java/codeconv-138413.html>

(TODO: *.tex Dateien)

4.2.2 Kommentarsprache

Die Sprache in der unsere Kommentare verfasst sind, wird Deutsch sein. Dies verhindert mögliche Missverständnisse innerhalb unserer Gruppe.

4.2.3 JavaDoc

Wir benutzen JavaDoc zur Dokumentation unseres Quellcodes. Dieses lässt eine einfache Erstellung von HTML-Dokumentationsdateien zu. Zur zukünftigen Wartung wird bis auf triviale Codezeilen, der komplette Code in JavaDoc dokumentiert.

4.2.4 Begleitende Dokumentation

(TODO: Weitere Erläuterung zur Doku?)

4.3 Unterstützende Projektfunktionen

In Abschnitt (TODO: Abschnitt nennen) werden unsere Phasenleiter benannt, die für die jeweiligen Phasen verantwortlich sind und als Ansprechpartner und Leiter dienen.

Unsere Projektdateien stehen jederzeit auf *GitHub*¹³ zur Verfügung. Außerdem werden regelmäßig von *jedem* Gruppenmitglied Datensicherungen, in Form von Updates und Backups des Repositorys auf dem eigenen Rechner stattfinden.

(TODO: Qualitätssicherung)

5 Arbeitspakete, Zeitplan und Budget

Dieser Teil ist ein zweiter Schwerpunkt des Projektplans. Hier sollt Ihr die nächste Phase detailliert planen (siehe Arbeitspakete). Die weiteren Phasen sollen ebenfalls wenigstens grob geplant werden. Ein Gantt-Diagramm ist zwingend!

Ihr sollt den Plan in der kommenden Phase auch tatsächlich benutzen – und so Erfahrungen sammeln, was evtl. bei der Planung unberücksichtigt blieb. Bei der nächsten Zeitplanung (für die nächste Phase) bekommt Ihr dann evtl. eine noch bessere Planung hin.

¹³asd

5.0.1 Anmerkungen und Annahmen

Wir haben bisher nur die Phasen Projektplan und Anforderungsspezifikation vollständig in den Arbeitspaketen und Zuteilungen behandelt. Die übrigen Phasen des Projekts - Architekturbeschreibung, Implementierung und Test - können zu diesem Zeitpunkt noch nicht detailliert beschrieben werden, sondern erfolgen stattdessen in grobem Format.

Arbeitspunkte zusammen

5.1 Arbeitspakete

Im Folgenden sind die einzelnen Arbeitspakete des Abschnitts Projektplan aufgeführt.

1.Arbeitspakete für Projektplan

Arbeitspaket 1.1

Bezeichnung	Vorbereitung für Projektplanerstellung
Beschreibung:	
Hauptverantwortlicher	Prüfsumme
Abhängigkeit
Ressourcen
Aufwand und Gesamtdauer	10h, 120h
Beginn
Ende
Mindestanforderungen:	

Arbeitspaket 1.2

Bezeichnung	Vorbereitung für Projektplanerstellung
Beschreibung:	
Hauptverantwortlicher	Prüfsumme
Abhängigkeit	...
Ressourcen	GitHub,...
Aufwand und Gesamtdauer	10h, 120h
Beginn	13.10.2013
Ende	20.10.2013
Mindestanforderungen:	

Besonderen Wert legen wir auf die Granularität der APs. Diese sollten von 1-2 Personen in max. einer Woche Zeitdauer (kalendarisch, nicht Aufwand) bearbeitbar sein. Die Beschreibungen sollten so genau sein, dass der Bearbeiter damit genau weiß, was zu tun ist.

5.2 Zeitplan und Abhängigkeiten

Die Abhängigkeiten zwischen Arbeitspaketen oder Meilensteinen müssen genannt werden, sowie im Gantt-Diagramm eingezeichnet werden. Der kritische Pfad soll angegeben und/oder eingezeichnet werden!

5.3 Ressourcenanforderung

Jedem Arbeitspaket muss mind. ein Bearbeiter zugeordnet werden. Die Zuordnung der ganzen Gruppe sollte nur in Ausnahmefällen erfolgen – und dann vermutlich begründet werden!

6 Sonstige Elemente

ENTFÄLLT

6.1 Pläne für die Konvertierung von Daten

ENTFÄLLT

6.2 Managementpläne für Unterauftragsnehmer

ENTFÄLLT *Wenn Fremdbibliotheken benutzt werden...*

6.3 Ausbildungspläne

ENTFÄLLT *Hierunter fallen z.B. auch interne Schulungen, die Ihr durchführen wollt.*

6.4 Raumpläne

ENTFÄLLT ...

6.5 Installationspläne

ENTFÄLLT ...

6.6 Pläne für die Übergabe des Systems

ENTFÄLLT ...